IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Appellants:	Jagadish Bandhole, Sekaran Nanja, Shan Balasubramaniam		
Assignee:	Symantec Operating Corporation		
Title:	Collaborative Computing Systems Using Dynamic Computing Environments		
Application No.:	09/888,110	Filing Date:	June 22, 2001
Examiner:	Kristie D. Shingles	Group Art Unit:	2141
Docket No.:	VRT0074US	Confirmation No.:	7964

Austin, Texas
September 8, 2008

Mail Stop: Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF UNDER 37 CFR § 41.37

Dear Sir:

This brief is timely submitted in support of the Notice of Appeal regarding the final rejection of claims 1-24. Appellants note that a Notice of Appeal and Pre-Appeal Brief Request for Review was filed May 7, 2008 (having a two-month extendable period for filing an Appeal Brief) , thereby giving the Appellant a period for filing set to expire on July 7, 2008. Filed herewith is a Petition for Extension of Time requesting a two-month extension, thereby giving the undersigned a period of September 8, 2008 in which to respond (September 7, 2008 having been a Sunday).

Please charge deposit account No. 502306 for the fee of $510.00 associated with this appeal brief. Please charge this deposit account for any additional sums which may be required to be paid as part of this appeal.

I. REAL PARTY IN INTEREST

The real party in interest on this appeal is Symantec Operating Corporation.

II. RELATED APPEALS AND INTERFERENCES

There are no appeals or interferences related to this application.

III. STATUS OF CLAIMS

Claims 1-24 are pending in the application.

Claims 1-24 stand rejected.

The Appellants appeal the rejection of claims 1-24.

IV. STATUS OF AMENDMENTS

No amendments were filed subsequent to the final rejection of February 7, 2008.

V. SUMMARY OF CLAIMED SUBJECT MATTER

Claim 1 describes a method for collaborative computing in a system. The method involves allocating a dynamic computing environment (e.g., DCE 104 of FIG. 1; further examples of a DCE are provided in lines 3-16 of page 9 of the specification) using a first user interface (e.g., such as one implemented on USER 1 PC of FIG. 1, as described in lines 3-9 of page 11 of the specification). The dynamic computing environment includes at least one of a plurality of resources (e.g., resources PC A, PC B, PC C, PC D, PC E, PC F, PC G, PC H, and S/W 102 of FIG. 1; such resources are also generally described in lines 5-18 of page 7 of the specification), and the dynamic computing environment is allocated by virtue of allocating the at least one resource (e.g., PC D of FIG. 1; such resources are generally described on lines 22-27 of page 9 of the specification).

The method then involves sharing the at least one resource (e.g., PC D of FIG. 1) between the first user interface (e.g., a user interface implemented on USER 1 PC; such a user interface is generally described in lines 16-18 of page 8) and a second user interface (e.g., a user interface implemented on USER 2 PC, as described in lines 1-3 of page 13 of the specification; such a user interface is also generally described in lines 16-18 of page 8). An application (e.g., software (S/W) process 102 of FIG. 1) is executed on the at least one resource (e.g., PC D) using either the first user interface or the second user interface (e.g., as described in lines 1-3 of page 11 of the specification). The first user interface and the second user interface are at least in part provided by software executing on respective first (e.g., USER 1 PC of FIG. 1) and second devices (e.g., USER 2 PC of FIG. 1) separate from the dynamic computing environment (e.g., as shown in FIG. 1, USER 1 PC and USER 2 PC are separate from DCE 104).

Information generated by execution of the application (e.g., S/W 102 of FIG. 1) is transferred to the first user interface (e.g., as implemented on USER 1 PC of FIG. 1 and as described in lines 16-23 of page 11 of the specification). The information generated by execution of the application is also transferred to the second user interface (e.g., as implemented on USER 2 PC of FIG. 1 and as described in lines 16-23 of page 11 of the specification) in response to a command to collaborate (e.g., as described in lines 7-9 of page 6, lines 19-31 of page 11, and lines 14-16 of page 13 of the specification) with the second user interface.

Claim 5 sets forth a method for providing sharing of a software process (e.g., S/W process 102 of FIG. 1) among multiple users (e.g., USER1 and USER2 of FIG. 1). The method involves allocating a distributed computing environment (e.g., DCE 104 of FIG. 1) by virtue of allocating (e.g., as described in lines 14-24 of page 12 of the specification) a first user computer (e.g., USER 1 PC and/or one of PCs A-H of FIG. 1) and a second user computer (e.g., USER 2 PC and/or one of PCs A-H of FIG. 1). The method also involves using a resource computer (e.g., PC D of FIG. 1) to transmit information about execution of the process (e.g., S/W process 102 of FIG. 1) to the first user computer. The resource computer (e.g., PC D of FIG. 1) executes the process in a first location (e.g., the location of DCE 104), and a first user operates the first user computer in a second location (e.g., USER 1 can operate USER 1 PC and/or a first user computer allocated in the DCE from the location of USER 1 PC, as shown in FIG. 1 and as described in lines 31-33 of page 12 of the specification).

The method also involves using the resource computer (e.g., PC D of FIG. 1) to transmit information about the execution of the process (e.g., S/W process 102 of FIG. 1) to the second user computer (e.g., USER 2 PC and/or one of PCs A-H of FIG. 1). A second user (e.g., USER 2) operates the second user computer in a third location (e.g., USER 2 can operate USER 2 PC and/or a first user computer allocated in the DCE from the location of USER 2 PC, as shown in FIG. 1 and as described in lines 31-33 of page 12 of the specification). The first user computer and the second user computer comprise the distributed computing environment.

Claim 18 sets forth a system for sharing a software process (e.g., S/W process 102 of FIG. 1) among multiple users (e.g., USER1 and USER2 of FIG. 1). The system includes a resource computer (e.g., PC D of FIG. 1) that executes the process and transmits information about the process. The system also includes a first user computer (e.g., USER 1 PC) in a second location (e.g., the location of USER 1 PC) configured to receive information about the execution of the process. The system includes a second user computer (e.g., USER 2 PC) in a third location (e.g., the location of USER 2 PC) configured to receive information about the execution of the process. The system also includes a dynamic computing environment (e.g., DCE 104 of FIG. 1). The resource computer is allocated to allocate at least a portion of the dynamic computing environment

(e.g., allocating or provisioning a DCE involves allocating a resource, such as PC D of FIG. 1, in the DCE; such resources are generally described on lines 22-27 of page 9 of the specification).

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

I. Whether claims 1-4 are patentable under 35 U.S.C. §103(a) over Raja et al. (USPN 7,058,947) (hereinafter referred to as "Raja") in view of VMware, Inc., Technical White Paper, February 1999 (hereinafter referred to as "VMware").

II. Whether claims 5-14 and 18-21 are patentable under 35 U.S.C. §103(a) over VMware in view of McNally et al. (USPN 6,259,448) (hereinafter referred to as "McNally").

III. Whether claims 15-17 and 22-24 are patentable under 35 U.S.C. §103(a) over VMware in view of McNally and further in view of Ansberry et al. (USPN 5,887,170) (hereinafter referred to as "Ansberry").

VII. ARGUMENT

I. Claims 1-4 are patentable under 35 U.S.C. §103(a)

Claim 1 recites: "sharing the at least one resource between the first user interface and a second user interface," "transferring information generated by execution of the application to the first user interface," and "transferring the information generated by execution of the application to the second user interface in response to a command to collaborate with the second user interface, wherein the first user interface and the second user interface are at least in part provided by software executing on respective first and second devices separate from the dynamic computing environment." The Final Office Action mailed February 7, 2008 (hereinafter referred to as "FOA") notes that Raja fails to teach these features, and thus relies solely upon VMware to teach these features of claim 1. FOA, p. 4.

VMware, both alone and in combination with the cited portions of Raja, fails to teach or suggest sharing a resource between a first and a second user interface. None of the cited portions of VMware mention anything about multiple user interfaces, or about sharing a resource between multiple user interfaces. At best, VMware states that virtual machines can share files and devices (VMware, p. 1, paragraph 3) and that the application portion of Virtual Platform acts like a normal application to use the graphical user interface of the host operating system to administer virtual machines (VMware, p. 4, paragraph 2).

Furthermore, the host operating system's user interface (mentioned on page 4 of VMware) is the only user interface mentioned in the cited portions of VMware, and thus the reference clearly only discloses a single user interface. As such, the reference clearly does not teach or suggest sharing a resource among multiple user interfaces, since only a single user interface is disclosed.

In response to the above arguments, the FOA asserts that "VMware clearly teaches users sharing resources between multiple interfaces of virtual machines by allowing multiple operating systems to run concurrently using the same hardware resources; which allows for the virtual machines to share files and devices." FOA, p. 2.

However, it is irrelevant that the virtual machines share files and devices, since the virtual machines are not user interfaces of the type recited in claim 1.

Furthermore, if the virtual machines run concurrently using the same hardware resources, as alleged in the FOA, then the virtual machines are clearly not "at least in part provided by software executing on respective first and second devices separate from the dynamic computing environment." In other words, the virtual machines run on the same hardware resources, not on different (first and second) devices in the manner of the user interfaces of claim 1.

For a similar reason, the cited portions of VMware, both alone and in combination with the cited portions of Raja, fail to teach or suggest that the first user interface and the second user interface are at least in part provided by software executing on respective first and second devices separate from the dynamic computing environment. As noted above, only a single user interface is disclosed in the cited portions of VMware. Furthermore, the cited portions of VMware only disclose a single device. Figures 1 and 3 clearly show only a single device running the VMware Virtual Platform (TM), and Figure 2 appears to simply illustrate different configurations of a single machine running Virtual Platform. Thus, VMware shows only a single device and does not describe how different user interfaces could be implemented on different devices in a manner consistent with that recited in claim 1.

Finally, VMware, both alone and in combination with the cited portions of Raja, fails to teach or suggest transferring information generated by execution of the application to the second user interface in response to a command to collaborate with the second user interface. Nothing in the cited portion of VMware describes or suggests a command to collaborate, nor does the cited portion of VMware suggest performing any action, let alone the action of transferring information generated by executing an application, in response to such a command.

In response to the above arguments, the FOA restates the assertion that "VMware clearly teaches allowing multiple operating system environments to run concurrently using the same hardware resources wherein virtual machines are allowed to share files... which clearly implies the transferring of information from programs running in one virtual machine environment to another virtual machine environment." FOA, p. 2.

However, this ignores several features of claim 1. First, claim 1 recites that information is transferred to a user interface, not to a virtual machine. Accordingly, it is irrelevant that VMware teaches transferring information between virtual machines, since those virtual machines are not user interfaces. Secondly, as noted above, the user interfaces of claim 1 are implemented on different devices, while the virtual machines of VMware appear to run on the same hardware.

Finally, claim 1 states that the information is transferred in response to a command to collaborate. No such command is disclosed in either reference, and the FOA does not cite any section of either reference as teaching such a command. Accordingly, without teaching such a command, and the transferring of information in response to such a command, the combined references clearly fail to teach or suggest the features of claim 1.

Accordingly, claim 1 is patentable over the cited art for at least the foregoing reasons. Dependent claims 2-4 are patentable over the cited art for similar reasons.

II. Claims 5-14 and 18-21 are patentable under 35 U.S.C. §103(a)

With respect to claim 5, the cited art fails to teach or suggest "using a resource computer to transmit information about execution of the process to the first user computer, wherein the resource computer executes the process in a first location, and a first user operates the first user computer in a second location; and using the resource computer to transmit information about the execution of the process to the second user computer, wherein a second user operates the second user computer in a third location, and the first user computer and the second user computer comprise the distributed computing environment."

The Office Action relies upon pages 4 and 5 of VMware to teach these features of claim 5. Office Action, p. 4. However, these portions of the reference merely describe how the components of the Virtual Platform interact (page 4) and that the Virtual Platform can be used to assign various I/O devices to virtual machines (page 5). In particular, the VMware Virtual Platform (TM) includes an application, which executes on top of the operating system, a monitor, which executes beneath the operating system, and

a driver, which is part of the operating system and facilitates communication between the application and monitor. VMware, page 4.

Nothing in the cited portion of VMware teaches or suggests transmitting information about the execution of a process on one computer (or even one virtual machine) to multiple other computers (or other virtual machines). At best, the reference says: "During execution, the monitor calls back to the application to access system resources. The application then calls the host operating system to access these resources." VMware, page 4, paragraph 2. As shown in Figure 3 of VMware, the Virtual Platform components do not execute on different machines (virtual or otherwise); instead the Virtual Platform (VP) applications and monitor execute on the same device and are independent of the virtual machines implemented on that device. Thus, VMware clearly does not teach or suggest transmitting information about the execution of a process from one computer or virtual machine to another; instead, it merely describes two software components on the same device communicating with each other in order to support one or more virtual machines.

For at least this reason, the cited art fails to teach or suggest using a resource computer to transmit information about execution of the process to the first user computer and using the resource computer to transmit information about the execution of the process to the second user computer. These features is also not taught or suggested in McNally (which is not relied upon in the rejection of claim 5).

In response to the above arguments, the FOA cites p. 2 of VMWare as teaching "transferring data of an 'entire computing environment' between computers." FOA, p. 3. The cited section of VMWare states: "Encapsulate an entire computing environment and move it between computers as easily as copying a file." This clearly does not teach the scenario of claim 5, which involves transferring information about the execution of a process on one computer from the computer on which the process is executing to two other user computers. The "entire computing environment" does not appear to be information about the execution of a process, nor is the entire computing environment transferred in the specific manner recited in claim 5. Accordingly, claim 5 and its dependent claims 6-14, which are grouped with claim 5 for purposes of this appeal, are patentable over the cited art for at least the foregoing reasons. Claims 18-21, which are

also grouped with claim 5 for purposes of this appeal, are patentable over the cited art for similar reasons.

III. Claims 15-17 and 22-24 are patentable under 35 U.S.C. §103(a)

Dependent claims 15-17 and 22-24 stand rejected under 35 U.S.C. §103(a) as being unpatentable over VMware in view of McNally and further in view of Ansberry et al. (USPN 5,887,170) ("Ansberry"). The Appellants respectfully traverse these rejections for at least the reasons set forth above with respect to claims 5 and 18. Accordingly, these claims are grouped with claim 5 for purposes of this appeal.

CONCLUSION

The appellants respectfully submit that claims 1-24 are allowable over the cited references for at least the above-stated reasons. The appellants respectfully request that the Board reverse the rejections of these claims.

Respectfully submitted,

/Brenna A. Brock/

Brenna A. Brock
Attorney for Appellant
Reg. No. 48,509
512-439-5087
512-439-5099 (fax)

VIII. CLAIMS APPENDIX

1. (Previously Presented) A method for collaborative computing in a system, the method comprising:

allocating a dynamic computing environment using a first user interface, wherein the dynamic computing environment comprises at least one resource of a plurality of resources, and the dynamic computing environment is allocated by virtue of allocating the at least one resource;

sharing the at least one resource between the first user interface and a second user interface;

executing an application on the at least one resource using either the first user interface or the second user interface;

transferring information generated by execution of the application to the first user interface; and

transferring the information generated by execution of the application to the second user interface in response to a command to collaborate with the second user interface, wherein the first user interface and the second user interface are at least in part provided by software executing on respective first and second devices separate from the dynamic computing environment.

2. (Original) The method of claim 1, further comprising modifying the information in the first user interface by interacting with the at least one shared resource through the first user interface.

3. (Original) The method of claim 1, further comprising modifying the information in the second user interface by interacting with the at least one shared resource through the second user interface.

4. (Original) The method of claim 1, further comprising switching control to modify the information between the first and second user interface.

5. (Previously Presented) A method for providing sharing of a software process among multiple users, the method comprising:

allocating a distributed computing environment by virtue of allocating a first user computer and a second user computer;

using a resource computer to transmit information about execution of the process to the first user computer, wherein the resource computer executes the process in a first location, and a first user operates the first user computer in a second location; and

using the resource computer to transmit information about the execution of the process to the second user computer, wherein a second user operates the second user computer in a third location, and the first user computer and the second user computer comprise the distributed computing environment.

6. (Original) The method of claim 5, further comprising controlling the resource computer with the first user computer.

7. (Original) The method of claim 5, further comprising controlling the resource computer with the second user computer.

8. (Original) The method of claim 5, further comprising switching control of the resource computer between the first and second user computers.

9. (Original) The method of claim 5, further comprising modifying the information using the first user computer.

10. (Original) The method of claim 5, further comprising modifying the information using the second user computer.

11. (Original) The method of claim 5, further comprising switching control to modify the information between the first and second user computer.

12. (Original) The method of claim 5, wherein the shared software process is an operating system.

13. (Original) The method of claim 5, wherein the shared software process is a user interface controller.

14. (Original) The method of claim 5, further providing for sharing of a plurality of software processes.

15. (Original) The method of claim 5, wherein the system is used in training.

16. (Original) The method of claim 5, wherein the system is used in technical support.

17. (Original) The method of claim 5, wherein the system is used in usability studies.

18. (Previously Presented) A system for sharing a software process among multiple users, the system comprising:

a resource computer that executes the process and transmits information about the process;

a first user computer in a second location configured to receive information about the execution of the process;

a second user computer in a third location configured to receive information about the execution of the process; and

a dynamic computing environment, wherein the resource computer is allocated to allocate at least a portion of the dynamic computing environment.

19. (Original) The system of claim 18, wherein the dynamic computing environment is remotely located from the second and third location.

20. (Original) The system of claim 18, wherein the second location is remotely located from the third location.

21. (Original) The system of claim 18, further comprising a user interface controller, wherein the user interface controller switches control of the resource computer from the first user computer to the second user computer.

22. (Original) The system of claim 18, wherein the system is used in training.

23. (Original) The system of claim 18, wherein the system is used in technical support.

24. (Original) The system of claim 18, wherein the system is used in usability studies.

IX. EVIDENCE APPENDIX

None

X. RELATED PROCEEDINGS APPENDIX

None